Calum Crawford
9/11/2024
Foundations of AI

Algorithm	Time Complexity	Space Complexity	Complete?	Optimal?
BFS	O(b^d+1)	O(b^d+1)	Y	Y - assuming that all actions have the same cost
DFS	O(b^n)	O(bn)	Y	N - there is no guarantee that the first solution will be optimal
DLS	O(bd)	O(bd)	Y	N - if there are numerous solutions to the problem
IDS	O(b^d)	O(b^d)	Y	Y - only if the costs of the edge are equal to each other
A*	O(b^d)	O(b^d)	Y	Y - when the heuristic computed is valid
UCS	O(b^d)	O(b^d)	Y	Y

Total State Counts

Before Problem Decomposition

BFS: 66
DFS: 27
DLS: 27
IDS: 112

After Problem Decomposition

BFS: 20
DFS: 15
DLS: 15
IDS: 34

A* With an SLD for the heuristic: 61 States *if starting from position 8,8 (will be less than UCS if starting at say 4,4)
Uniform Cost Search: 61 States *if starting from position 8,8

Question 4. Responses

a) What were the engineering advances that led to Deep Blue's success? Which of them can be transferred to other problems, and which are specific to chess?

Some of the engineering advances that led to Deep Blue's success include a single-chip chess search engine, a massively parallel system with multiple levels of parallelism, strong emphasis on search extensions, a complex evaluation system, and an effective use of a Grandmaster game database. Some of the engineering features that could be transferred to other problems include a massively parallel system with multiple levels of parallelism. This could be helpful for numerous machine learning tasks including self-driving cars which need to perform numerous computations in parallel. Another feature could include a complex evaluation system. Certain problems such as classifying images require complex evaluation systems in order to distinguish between images. Some of the features that are specific to chess include a single-chip chess search engine, and the deep integration of the Grandmaster game database.

b) AlphaZero is compared to a number of modern game-playing programs, such as StockFish, which work similarly to Deep Blue. The paper shows that AlphaZero is able to defeat StockFish even when it is given only 1/100 of the computing time. Why is that? Please frame your answer in terms of search and the number of nodes evaluated.

AlphaZero is able to defeat StockFish even when it is given only 1/100 of the computing time because it doesn't have to evaluate as many nodes. In the context of chess, Stockfish analyzes 70 million positions per second compared to only 80,000 computed by AlphaZero. AlphaZero is able to do this by using its deep neural network to focus more selectively on the most promising variations/moves. The paper argues this is a more "human-like" approach to search, it doesn't need to evaluate every single possible combination, only the ones that are likely to lead to a win.